

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Elektronický zpěvník

Electronic songbook

Zadání bakalářské práce

Student:

Štěpán Drozdek

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R025 Informatika a výpočetní technika

Téma:

**Elektronický zpěvník
Electronic Songbook**

Jazyk vypracování:

čeština

Zásady pro vypracování:

Cílem práce je vytvořit webový informační systém, takzvaný elektronický zpěvník, který bude sloužit hudebníkům, skladatelům a kapelám ke správě, organizaci a prezentaci textu a akordů písní pro kytary. Systém bude umožňovat vkládání a editaci textu a akordů písní, jejich import z jiných formátů a také synchronizované zobrazení skladby pro několik zařízení.

Úkoly:

1. Prostudovat stávající elektronické zpěvníky a zhodnotit jejich vlastnosti.
2. Navrhnout a naimplementovat webovou aplikaci, která bude obsahovat nástroje pro editaci textu písní a akordů. Aplikace bude obsahovat nástroje pro import textu a akordů z jiných formátů.
3. Navrhnout vhodné datové struktury pro persistenci dat v lokální a centrální databázi.
4. Navrhnout a naimplementovat webovou aplikaci, která bude vhodným způsobem prezentovat uložená data (zpěvníky). Aplikace bude obsahovat automatické posouvání textu a synchronizaci posouvání pro několik zařízení.
5. Srovnat aplikaci s existujícími systémy.

Seznam doporučené odborné literatury:

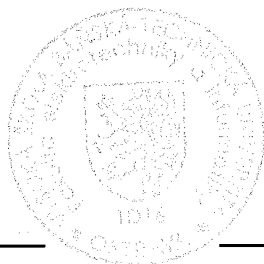
- [1] Adam Freeman. Pro ASP.NET MVC 5 (Expert's Voice in ASP.Net) 5th ed. Edition.
- [2] Asim Hussain. Angular 4: From Theory To Practice: Build the web applications of tomorrow using the new Angular web framework from Google.


Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.


Vedoucí bakalářské práce: **Ing. Peter Chovanec, Ph.D.**

Datum zadání: 01.09.2017

Datum odevzdání: 30.04.2018

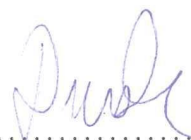



doc. Ing. Jan Platoš, Ph.D.
vedoucí katedry


prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 30. dubna 2018



.....

Děkuji tímto vedoucímu bakalářské práce Ing. Petrovi Chovancovi, Ph.D. za odbornou pomoc, ochotu a konzultace při realizaci tohoto projektu.

Abstrakt

Cílem práce je vytvořit webový informační systém, takzvaný elektronický zpěvník, který bude sloužit hudebníkům, skladatelům a kapelám ke správě, organizaci a prezentaci textu a akordů písní pro kytary. Systém bude umožňovat vkládání a editaci textu a akordů písní, jejich import z jiných formátů a také synchronizované zobrazení skladby pro několik zařízení.

Klíčová slova: Informační systém, Elektronický zpěvník, Návrh doménového modelu, ASP.NET Core 2.0, Angular 4, MS SQL, Typescript.

Abstract

The aim of the thesis is to create a web-based information system, so-called electronic songbook, which will serve musicians, composers and bands to manage, organize and present text and song chords for guitars. The system will allow you to insert and edit text and song chords, to import them from other formats, and to synchronize track views for multiple devices.

Key Words: Information system, Electronic songbook, Domain mode design, ASP.NET Core 2.0, Angular4, MS SQL, Typescript.

Obsah

Seznam použitých zkratk a symbolů	7
Seznam obrázků	8
Seznam tabulek	9
Seznam výpisů zdrojového kódu	10
1 Úvod	11
2 Formáty hudebních děl	12
2.1 ABC Music Notation	12
2.2 JAM Notation	12
2.3 Lily Pond	13
2.4 ChordPro	13
2.5 Textový soubor	14
3 Analýza systému	15
3.1 Vize	15
3.2 Datová analýza	15
3.3 Funkční analýza	20
4 Použité technologie	23
4.1 Datová vrstva	23
4.2 Aplikační vrstva	23
4.3 Prezentační vrstva	24
5 Implementace	26
5.1 Základní popis	26
5.2 Autentizace a správa uživatelů	28
5.3 Vložení nové písně	28
5.4 Vizualizace písně	30
5.5 Synchronizované skupinové hraní	33
6 Porovnání s existujícími systémy	39
7 Závěr	41
Literatura	42

Seznam použitých zkratk a symbolů

IS	– Informační Systém
ORM	– Objektově Relační Mapování
UML	– Unified Modeling Language
HTML	– Hyper Text Markup Language
CSS	– Cascading Style Sheets
JS	– JavaScript

Seznam obrázků

1	Formát hudebního díla - ABC Music Notation	12
2	Formát hudebního díla - JAM Notation	13
3	Formát hudebního díla - Lily Pond	13
4	Formát hudebního díla - ChordPro - Ukázka 1	14
5	Formát hudebního díla - ChordPro - Ukázka 2	14
6	ER diagram	17
7	Diagram užití - Nepřihlášený uživatel	20
8	Diagram užití - Přihlášený uživatel	21
9	Zobrazení stránky - Přihlášení a registrace	26
10	Zobrazení stránky - Nepřihlášený uživatel	27
11	Zobrazení stránky - Přihlášený uživatel	27
12	Zobrazení stránky - Přidání písně	29
13	Zobrazení stránky - Výběr písně pro synchronizované hraní	37
14	Zobrazení stránky - Ukázka zobrazené písně a chatu v synchronizovaném hraní .	38

Seznam tabulek

1	Schéma tabulky User	18
2	Schéma tabulky User_in_group	18
3	Schéma tabulky Group	18
4	Schéma tabulky Band_songs	18
5	Schéma tabulky Evaluation	18
6	Schéma tabulky Favorite_songs	19
7	Schéma tabulky Comments	19
8	Schéma tabulky Song	19
9	Schéma tabulky Band	19
10	Schéma tabulky Genre	19
11	Schéma tabulky Songs_genre	20

Seznam výpisů zdrojového kódu

1	Ukázka SignalR .NET kódu	24
2	Ukázka Angular 4 atributů v HTML kódu	25
3	Ukázka šifrování hesla	28
4	Rozložení dat ze souboru při vkládání písničky	29
5	Dělení textu - první část	30
6	Dělení textu - druhá část	31
7	Dělení textu - třetí část	32
8	InnerHTML	33
9	Automatický posun obrazovky	33
10	Vytvoření skupiny pro synchronizované hraní	35
11	Zavolání funkce pro vytvoření SignalR skupiny	35
12	Zavolání funkce pro připojení k SignalR skupině	35
13	Připojení ke skupině pro synchronizované hraní	36
14	Posílání zpráv všem členům skupiny	36
15	Listener na zprávy v synchronizovaném hraní	36

1 Úvod

S nástupem nových technologií elektronická zařízení nahrazují nebo rozšiřují papírové knihy a učebnice. Jejich víceúčelnost a skladnost je nepopíratelnou výhodou. A tak jako učebnice jsou postupně nahrazovány textovými soubory, tak i hudební zpěvníky mají možnost být převedeny do elektronické podoby. Jednou z největších nevýhod papírových zpěvníků je jejich omezená kapacita, kterou však elektronické zpěvníky nesdílejí. Ty navíc umožňují služby, které papírový zpěvník není schopen poskytnout.

Cílem této bakalářské práce je vypracovat nový elektronický zpěvník, který bude umožňovat přístup nejen přihlášeným uživatelům, ale i uživatelům, kteří chtějí využívat IS bez registrace a přihlašování. Nepřihlášení uživatelé však budou omezeni ve využívání služeb, které jsou přístupny pouze po přihlášení. IS bude vytvořen s responzivním designem, čímž uživatelé nebudou limitováni pouze na stolní počítače a notebooky, ale budou moci využívat i kapesních zařízení, jako jsou chytré telefony a tablety. S rozšiřující se popularitou mobilních dat bude uživatel schopen využívat služeb IS i na cestách bez nutnosti tisku či stahování písní. Díky synchronizovanému hraní budou moci uživatelé hrát ve větší skupině stejnou píseň se zobrazením na různých zařízeních.

Kvůli neustále se vyvíjející technologii je potřeba obnovovat a upravovat IS tak, aby odpovídaly stále rostoucím požadavkům uživatelů. Elektronický zpěvník, jenž byl vytvořen před 10 lety, nyní jen stěží bude splňovat vše, co od něj uživatel očekává. Cílem je vytvořit nový elektronický zpěvník, využívající nové technologie s možnostmi dalšího rozšiřování, který by více splňoval požadavky současných uživatelů. Elektronické zpěvníky, které již jsou na trhu, často neposkytují požadovanou funkcionalitu. Většina z nich nemá responzivní design, ani možnost synchronizovaného skupinového hraní. Mnohé neumožňují uživatelům vkládat písně, tudíž je IS zcela odkázán na práci administrátora, který tuto činnost vykoná.

2 Formáty hudebních děl

Existuje mnoho formátů hudebních děl, ale ne všechny jsou použitelné pro tento elektronický zpěvník. Aby se vybral správný formát pro vložení písně, je potřeba prostudovat, které formáty jsou vhodné.

2.1 ABC Music Notation

ABC Music Notation [1] music notation je formát, který nevyhovuje požadavkům IS. Umožňuje sice správu akordů, nikoliv však textu. Programy které využívají ABC music notation, převádějí text souboru do PNG souboru.

```
X: 3
T: Country Garden(s)
Z: 1997 by John Chambers <jc@trillian.mit.edu> http://trillian.mit.edu/~jc/music/abc/
P: A(A2B2C2B2)2
M: C|
L: 1/8
K: C
P: A
| "G7" g2 g>f "C" e2 e2 | "D7" d2 d>c "G" B2 B>c | "G" d2 G2 "C" A2 c2 | "D7" B3A "G" G4 ||
P: B
[| "A7" g>a g>e "D" f2 d2 | "A7" g>a g>e "D" f4 | "A7" g2g>f e2a2 | f3e "D7" d2B>c |
| "G7" d2 g>f "C" e2 e2 | "D7" d>e d>c "G" B2 B>c | "G" d2 G2 "C" A2 c2 | "D7" B3A "G" G4 ||
P: C
[| "C" G2c2 c2e2 | "D7" d>e d>c "G" B2 B>c | "G" d2 G2 "C" A2 c2 | "D7" B3A "G" G4 ||
```



Obrázek 1: Formát hudebního díla - ABC Music Notation

2.2 JAM Notation

Tak jako ABC Music Notation, i JAM Notation [2] slouží ke správě akordů, nikoliv textu. Způsob zápisu akordů se liší, avšak výsledné PNG zobrazuje noty stejně jako ABC Music Notation.

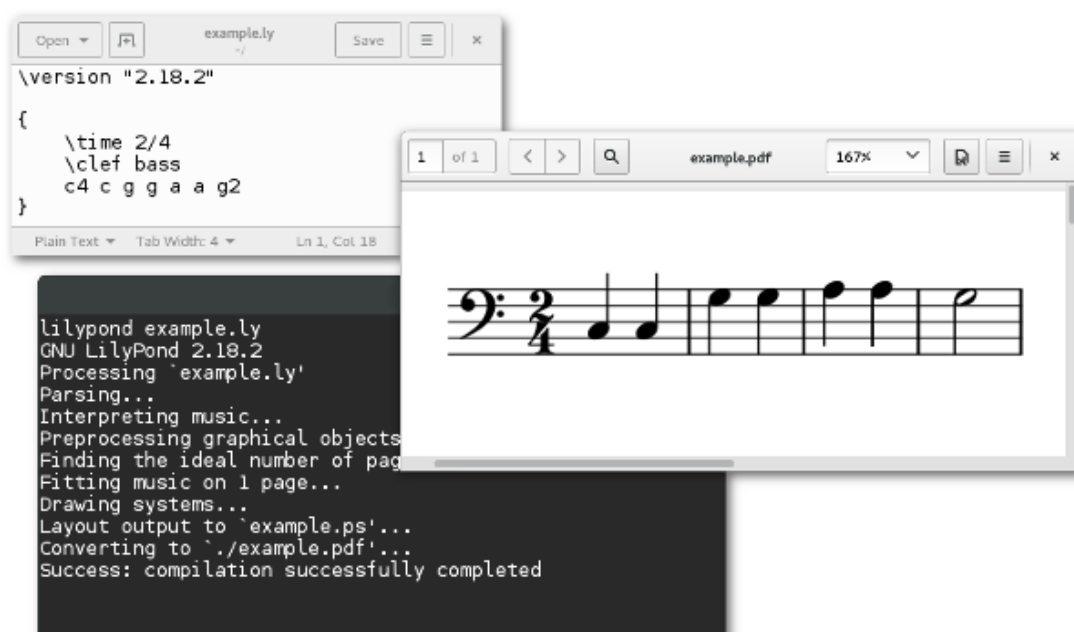
C== C7== /= | F== G7== /= | C /= ----



Obrázek 2: Formát hudebního díla - JAM Notation

2.3 Lily Pond

Lily Pond [3] slouží k převedení textu v určité syntaxi do notové osnovy. Bezpochyby velmi užitečný nástroj, avšak pro tento IS je nepoužitelný.



Obrázek 3: Formát hudebního díla - Lily Pond

2.4 ChordPro

ChordPro [4] je textový formát pro správu písní s akordy a textem. Tím je ideálním formátem pro náš IS. Umožňuje nejen správu akordů pro kytaru, ale i pro ostatní strunné nástroje, jako ukulele, basová kytara nebo mandolína.

ChordPro umožňuje uživateli specifikovat informace o písni do složených závorek. Mezi tyto informace patří například název, umělec, skladatel, kdy byla píseň vytvořena, atd.

Down in the hol[C]low, [G] playing a ne[D7]w game [G] ➔

C G
Down in the hollow,
G D7 G
playing a new game

Obrázek 4: Formát hudebního díla - ChordPro - Ukázka 1

Na následujícím obrázku je viditelné použití specifikovaných informací ve složených závorkách. Jedná se o titul a podtitul.

Hurt
Rythm of song is : DD

```
{t:Hurt}
{st:Johnny Cash}

[C]What have I [F]become?[C]
my sweetest [G]friend
[C]everyone I [F]know
goes [C]away in the [G]end
[C]you could have it [F]all
[C]my empire of dirt
[G]I will let you [C]down
[F]I will make you [Am]hurt
```

➔

C F C
What have I become? my sweetest
G C F C
friend everyone I know goes away in
G C
the end you could have it
F C G
all my empire of dirt I will
C F
let you down I will make
Am
you hurt

Obrázek 5: Formát hudebního díla - ChordPro - Ukázka 2

2.5 Textový soubor

Mimo ChordPro bude tento IS schopen pracovat i s textovými soubory. Tyto soubory budou formátovány jako ChordPro soubory. To znamená, že akordy budou v hranatých závorkách a informace ve složených závorkách.

3 Analýza systému

3.1 Vize

Je zapotřebí zřídit IS, takzvaný Elektronický zpěvník. Systém bude určen pro dvě skupiny uživatelů. První skupinou budou přihlášení uživatelé a druhou skupinou budou nepřihlášení uživatelé. Přihlášení uživatelé mají, na rozdíl od nepřihlášených uživatelů, k dispozici větší počet služeb, které usnadní a zpříjemní používání IS. Přihlášený i nepřihlášený uživatel má možnost zobrazování textu a akordů písní s možností automatického posouvání textu na obrazovce v rychlosti, kterou si sám vybere. Dále vyhledání písně podle názvu, umělce nebo žánru.

Pro přihlášené uživatele dále bude umožňovat synchronizované hraní ve skupině, kde zakladatel skupiny vybere píseň i rychlost automatického posouvání textu pro celou skupinu. IS umožňuje pro přihlášené uživatele přidání písně do oblíbených, hodnocení a komentování písně, čímž poskytuje autorovi zpětnou vazbu.

Systém bude zcela soběstačný co se týká přidávání písní. Každý registrovaný uživatel bude moci přidat libovolný počet písní, a to jak vypsáním do komponenty ve webové aplikaci, tak vložením souboru. A to buď textového souboru (.txt), nebo souboru ve formátu chordpro (.chordpro), který je v současné době nejpoužívanější formát pro uložení písní s akordy. Uživatel bude schopen písně, jenž vložil, dále upravovat, nebo je odstranit.

3.2 Datová analýza

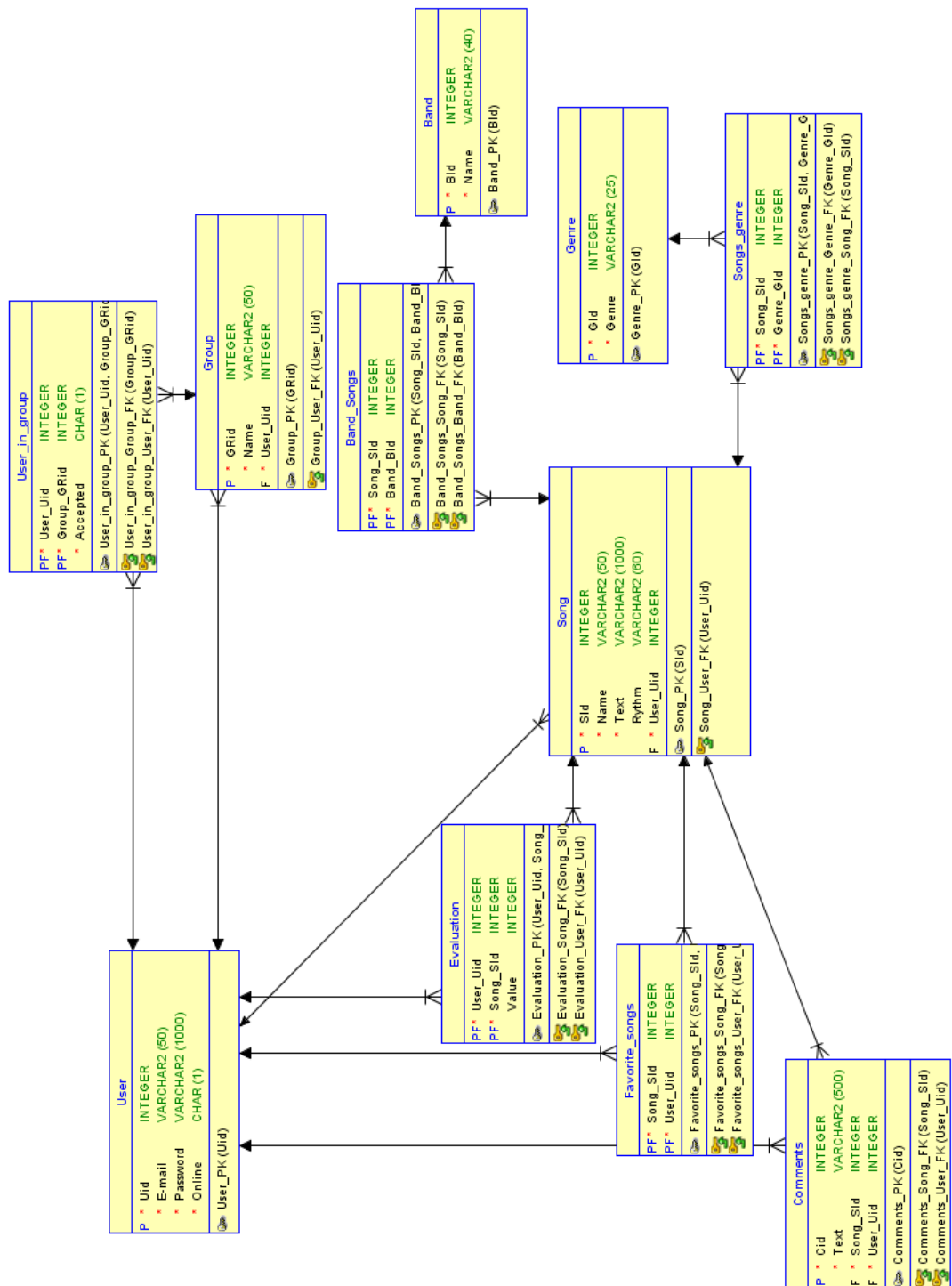
3.2.1 Vstupy do informačního systému

Vstupem jsou míněny informace, které uživatel zadává a jsou dále zpracovávány systémem a následně uloženy do databáze.

- Uživatel
 - Uživatel zadává své přihlašovací jméno a heslo. V případě registrace zadává heslo 2x z důvodu ověření překlíku.
- Píseň
 - Uživatel zadává název písně, jméno umělce a text písně. Žánr je vybírán z nabídky.
 - Uživatel má také možnost nahrát informace ze souboru, ty se zobrazí v inpuotech a uživatel je může dále upravovat.
- Skupina
 - Uživatel má možnost vytvořit skupinu. K vytvoření skupiny je potřeba, aby uživatel byl přihlášen a vybral jméno skupiny.

- Jakmile je skupina vytvořena, ostatní uživatelé mají možnost požádat o členství. To však musí být první schváleno zakladatelem skupiny.
- Umělec
 - Jméno umělce. Při vložení písničky je uživateli nabídnut seznam umělců. Za předpokladu, že v něm neexistuje jméno požadovaného umělce, zadaný text se uživatele uloží jako nový umělec.
- Hodnocení
 - Hodnocení v rozsahu 1-5 pro danou píseň.
 - U písničky se zobrazí průměrné hodnocení.
- Komentář
 - Komentář obsahuje automaticky vložené jméno uživatele a text, jenž uživatel napíše.
 - Maximální povolená velikost komentáře je 500 znaků, komentář delší než 500 znaků se neuloží.

3.2.2 Datový model



Obrázek 6: ER diagram

3.2.3 Datový slovník

Název	Typ	Velikost	Klíč	Null	Popis
UId	INTEGER		PK	Ne	Primární klíč uživatele
E-mail	VARCHAR2	50	Ne	Ne	Login uživatele
Password	VARCHAR2	1000	Ne	Ne	Heslo uživatele
Online	BOOLEAN		Ne	Ne	Určuje zda je uživatel online

Tabulka 1: Schéma tabulky User

Název	Typ	Velikost	Klíč	Null	Popis
User_UId	INTEGER		PFK	Ne	Primární klíč uživatele
Group_Grid	INTEGER		PFK	Ne	Primární klíč skupiny
Accepted	BOOLEAN		Ne	Ne	Informace zda je uživatel přijat do skupiny nebo ne

Tabulka 2: Schéma tabulky User_in_group

Název	Typ	Velikost	Klíč	Null	Popis
GRid	INTEGER		PK	Ne	Primární klíč skupiny
Name	VARCHAR2	50	Ne	Ne	Název skupiny
User_Uid	INTEGER		Ne	Ne	Primární klíč uživatele, jenž založil skupinu

Tabulka 3: Schéma tabulky Group

Název	Typ	Velikost	Klíč	Null	Popis
Song_SId	INTEGER		PFK	Ne	Primární klíč písně
Band_BId	INTEGER		PFK	Ne	Primární klíč umělce

Tabulka 4: Schéma tabulky Band_songs

Název	Typ	Velikost	Klíč	Null	Popis
User_Uid	INTEGER		PFK	Ne	Primární klíč uživatele, jenž vložil hodnocení
Song_SId	INTEGER		PFK	Ne	Primární klíč písně
Value	INTEGER		Ne	Ano	Hodnota daného hodnocení

Tabulka 5: Schéma tabulky Evaluation

Název	Typ	Velikost	Klíč	Null	Popis
Song_SId	INTEGER		PK	Ne	Primární klíč písně
User_UId	INTEGER		FK	Ne	Primární klíč uživatele

Tabulka 6: Schéma tabulky Favorite_songs

Název	Typ	Velikost	Klíč	Null	Popis
Cid	INTEGER		PK	Ne	Primární klíč komentáře
Text	VARCHAR2	500	Ne	Ne	Text komentáře
User_UId	INTEGER		FK	Ne	Cizí klíč uživatele, jenž vložil komentář
Song_SId	INTEGER		FK	Ne	Cizí klíč písně, ke které komentář patří

Tabulka 7: Schéma tabulky Comments

Název	Typ	Velikost	Klíč	Null	Popis
SId	INTEGER		PK	Ne	Primární klíč písně
Name	VARCHAR2	50	Ne	Ne	Název písně
Text	VARCHAR2	1000	Ne	Ne	Text písně
Rythm	VARCHAR2	60	Ne	Ano	Rytmus písně
User_UId	INTEGER		FK	Ne	Cizí klíč uživatele, jenž vložil píseň

Tabulka 8: Schéma tabulky Song

Název	Typ	Velikost	Klíč	Null	Popis
BId	INTEGER		PK	Ne	Primární klíč umělce
Name	VARCHAR2	40	Ne	Ne	Jméno umělce

Tabulka 9: Schéma tabulky Band

Název	Typ	Velikost	Klíč	Null	Popis
GId	INTEGER		PK	Ne	Primární klíč žánru
Genre	VARCHAR2	25	Ne	Ne	Název žánru

Tabulka 10: Schéma tabulky Genre

Název	Typ	Velikost	Klíč	Null	Popis
Genre_GId	INTEGER		PFK	Ne	Primární klíč žánru
Song_SId	INTEGER		PFK	Ne	Primární klíč písně

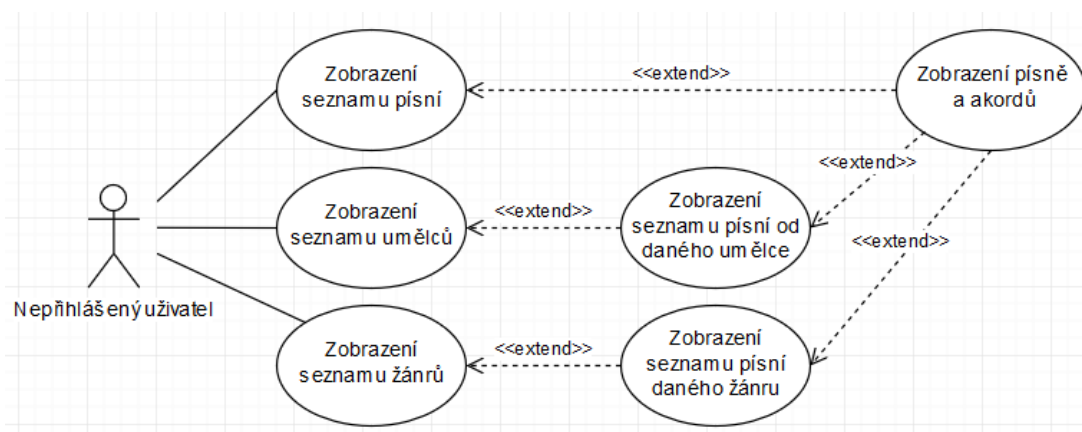
Tabulka 11: Schéma tabulky Songs_genre

3.3 Funkční analýza

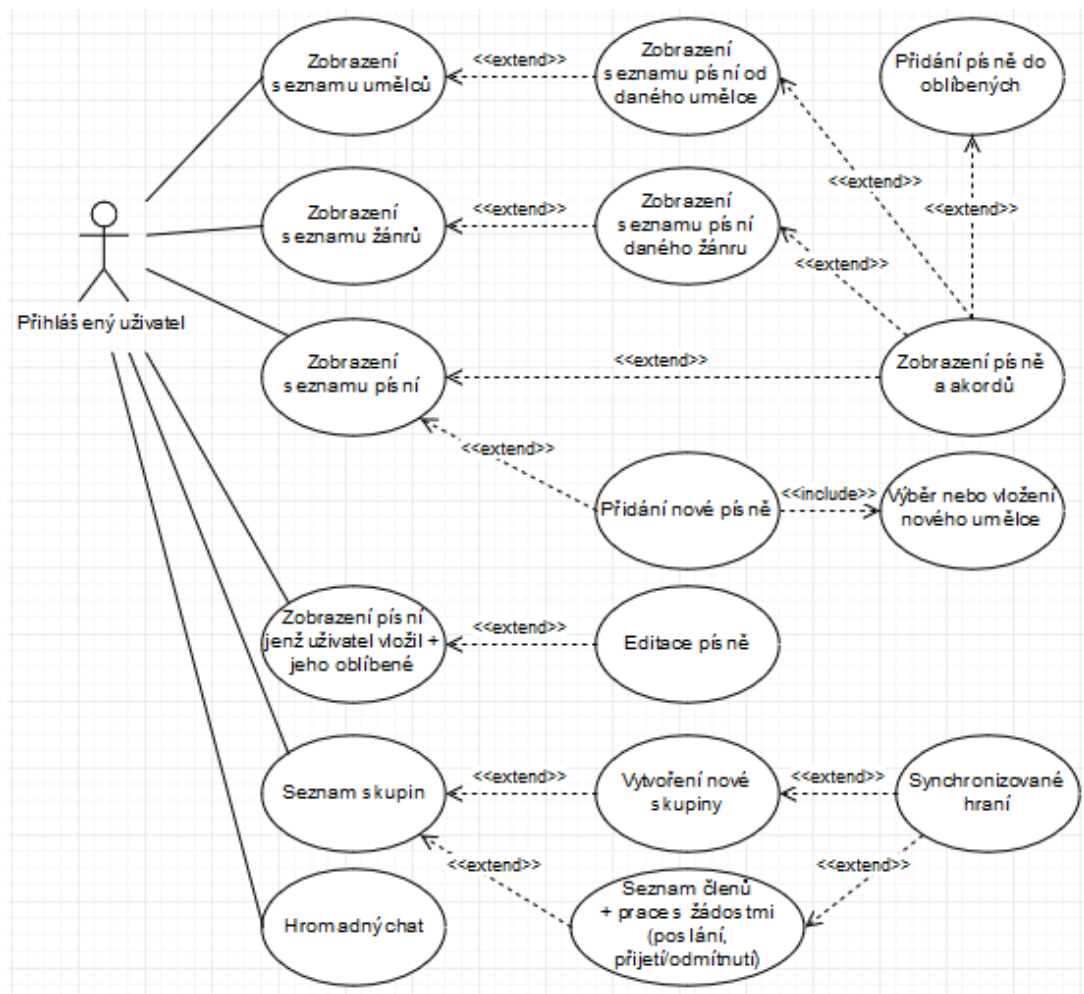
3.3.1 Kdo bude informační systém používat

Jak bylo řečeno v předchozí sekci, se systémem budou pracovat dvě skupiny, a to přihlášený uživatel a nepřihlášený uživatel.

- Nepřihlášený uživatel bude mít možnost vyhledávat v databázi informace, které požaduje, avšak nemůže žádným způsobem zasahovat (tzn. přidávat či upravovat) do dat uložených v databázi. IS mu umožní vyhledat písně, a to buď přes seznam písní, žánr ve kterém nalezne seznam všech písní daného žánru, nebo přes umělce jenž píseň vytvořil.
- Přihlášený uživatel bude mít k dispozici stejné služby jako nepřihlášený uživatel s tím rozdílem, že bude mít možnost zasahovat do dat uložených v databázi. Každý přihlášený uživatel bude moci přidávat písně, hodnotit či komentovat již uložené písně, vytvořit si seznam oblíbených písní nebo přidat nového umělce. V neposlední řadě budou mít přihlášení uživatelé možnost synchronizovaného skupinového hraní. K této činnosti musí být členy skupiny. Pokud je uživatel zakladatel skupiny, tak bude mít možnost vybrat jakou píseň bude skupina hrát a jakou rychlostí bude text automaticky posouván. Dále budou mít možnost se připojit k hromadnému chatu, ke kterému mají přístup všichni přihlášení uživatelé.



Obrázek 7: Diagram užití - Nepřihlášený uživatel



Obrázek 8: Diagram užití - Přihlášený uživatel

3.3.2 Seznam funkcí

- Správa uživatelů
 - Registrace
 - Přihlášení
 - Poslání žádosti o členství ve skupině
- Správa písni
 - Přidání písni
 - Výpis písni
 - Úprava písni
 - Odstranění písni

- Správa umělců
 - Přidání umělce
 - Vypis umělců
 - Výpis písní vybraného umělce
- Správa skupin
 - Vytvoření skupiny
 - Příjmutí žádosti o členství
 - Opuštění skupiny
 - Odstranění člena
- Správa hodnocení
 - Přidání hodnocení
 - Úprava hodnocení
 - Výpis hodnocení písně v seznamu písní
- Správa komentářů
 - Přidání komentáře
 - Výpis komentářů
- Správa oblíbených písní
 - Přidání písně do oblíbených
 - Odebrání písně z oblíbených
- Synchronizované skupinové hraní
 - Vytvoření nové skupiny
 - Přidání členů do skupiny
 - Posílání dat pro skupinu

3.3.3 Nefunkční požadavky

Výsledný IS by měl mít responzivní design, neboť se předpokládá využívání této aplikace na různých typech zařízení s různými zobrazovacími možnostmi. IS bude využívat nejmodernějších opensource technologií z důvodu efektivity vývoje a snižování nákladů. K těmto technologiím patří Angular4, který bude použit jako prezentační vrstva IS, ASP.NET sloužící jako aplikační vrstva, a MSSQL jakožto databáze.

4 Použité technologie

Před samotnou implementací je nutný výběr technologií, které budou použity při vývoji IS. Pro ukládání dat jsem se rozhodl využít MS SQL [8] databázový systém, jako aplikační vrstva bude sloužit ASP.NET Core 2.0 [10], ve kterém bude naimplementované ORM pro komunikaci s databází. Pro ORM využiji návrhových vzorů Data Mapper [6] a Table Data Gateway [7]. Data Mapper reprezentuje tabulky v databázi třídami, avšak bez CRUD operací, o ty se provádějí přes Table Data Gateway třídu. Pro prezentační vrstvu využiji Angular 4 [22], který je díky svým možnostem ideální pro tvorbu IS. Synchronizované skupinové hraní potřebuje komunikaci v reálném čase, proto využiji ASP.NET knihovnu SignalR [17], která tuto komunikaci umožňuje.

4.1 Datová vrstva

4.1.1 MS SQL

MS SQL [8] neboli Microsoft SQL Server je relační databázový a analytický systém vyvinutý společností Microsoft. Jeho primární funkcí je ukládání a načítání dat požadovaných jinými softwarovými aplikacemi. Tyto aplikace mohou běžet buď na stejném počítači nebo na jiném počítači v síti.

Společnost Microsoft uvedla a stále uvádí na trh velké množství verzí a edicí. Mezi verze patří například SQL Server 2008, SQL Server 2016, SQL Server 2017. Edice se dělí na 3 skupiny, a to jsou mainstream, specialized a discontinued, přičemž každá skupina obsahuje několik edicí dané verze.

MS SQL [8] také umožňuje vytvářet a upravovat uložené procedury, jenž jsou T-SQL [9] dotazy uložené na samotném serveru.

4.2 Aplikační vrstva

4.2.1 ASP.NET Core 2.0

ASP.NET Core 2.0 [10] je open-source multiplatformní framework k tvorbě webových a mobilních aplikací. Pro práci s .NET lze využít vývojové prostředí Visual Studio, nebo lze také tvořit z příkazového řádku.

Mimo své technické možnosti platforma nabízí škálu programovacích jazyků. Mezi ně patří C#, F# a Visual Basic. Tato práce bude psána v jazyku C#.

Verze 2.0 je oproti verzi 1.0 o přibližně 20% rychlejší a má přístup ke známým API, jako například Serializace [11], práci se sítí (Socket [12], HTTP [13], WebSocket [14]...), XML (XPath [15], XQuery [16]) a mnohé další.

4.2.2 SignalR

SignalR [17] je open-source knihovna technologie ASP.NET [10] pro komunikaci v reálném čase. To znamená, že uživatel nemusí posílat žádost o data, ale data jsou poslána automaticky. SignalR poskytuje jednoduché API pro vytvoření serveru a klienta. Vzdáleně volané procedury, neboli RPC, které volají JavaScriptové funkce na straně klienta z kódu .NET na straně serveru.

Nespornou výhodou SignalR [17] je, že má několik možností přenosu. A to jsou websocket [14], foreverFrame [18], serverSentEvents [19] a longPolling [20]. Který typ přenosu bude použit, rozhoduje SignalR sám, ale máme možnost sami specifikovat, který se má použít. V případě, že si typ nespecifikujeme sami a jeden z typů přenosu není podporován (například internetový prohlížeč Internet Explorer nepodporuje serverSentEvents), tak SignalR [17] automaticky přejde na jiný. Díky tomu můžeme mít jistotu, že data vždy dorazí na místo určení bez nutnosti implementace několika typu přenosu dat.

```
private void BroadcastGroup(PlayingGroups group)
{
    var clients = group.Members.Select(g => g.ConnectionId).ToList();
    Clients.Clients(clients).SendAsync("Group", group);
}
```

Výpis 1: Ukázka SignalR .NET kódu

4.3 Prezentační vrstva

4.3.1 HTML5

HTML5 [21] je značkový jazyk pro tvorbu webových stránek. HTML využívá tagů k definici struktury stránky. HTML5 se od HTML4 liší především podporou multimedií a možnosti podpory aplikací, které fungují bez připojení k internetu.

4.3.2 Angular 4

Angular 4 [22] je open-source platforma usnadňující vytváření webových aplikací. Kombinuje šablony, vkládání závislosti, end-tooling a best practices. Angular4 umožňuje vývojáři umístit do HTML stránky speciální atributy, díky kterým je stránka ovlivňována. Ať už díky oboustranné vazbě zobrazovaných proměnných, či díky atributům sloužícím jako například podmínky nebo cykly.

Angular 4 [23] narozdíl od Angular 2 a Angular 1 (neboli AngularJS), pracuje se současnou verzí TypeScriptu, což dělá aplikace menší a rychlejší. Také volání HTTP metod je zjednodušeno. Angular 4 má zpětnou kompatibilitu s Angularem 2, tudíž projekty vytvořené pomocí Angular 2 budou bez problému fungovat i v Angular 4.

Angular 4 [24] také nabízí velké množství balíčků usnadňujících práci. Díky tomuto se Angular stává ještě silnějším nástrojem, než už je. Vývojář není nucen řešit vše sám, ale má možnost si k činnosti stáhnout balíček, který mu může pomoci s řešením problému.

```
<table class='table' *ngIf="bands">
  <thead>
    <tr>
      <th>Name</th>
    </tr>
  </thead>
  <tbody>
    <tr *ngFor="let band of bands" (click)="goToBandSongs(band)">
      <td>{{ band.name }}</td>
    </tr>
  </tbody>
</table>
```

Výpis 2: Ukázka Angular 4 atributů v HTML kódu

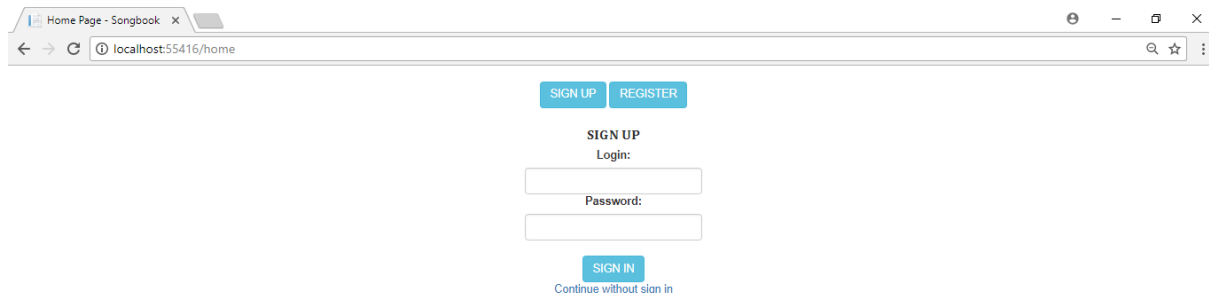
4.3.3 Bootstrap

Bootstrap [25] je open-source toolkit pro vývoj HTML, CSS a JS. Za pomoci rozsáhlých předinstalovaných komponentů a výkonných pluginů postavených na jQuery, můžeme stylovat a přidávat jednoduchou funkcionalitu naší HTML stránky.

5 Implementace

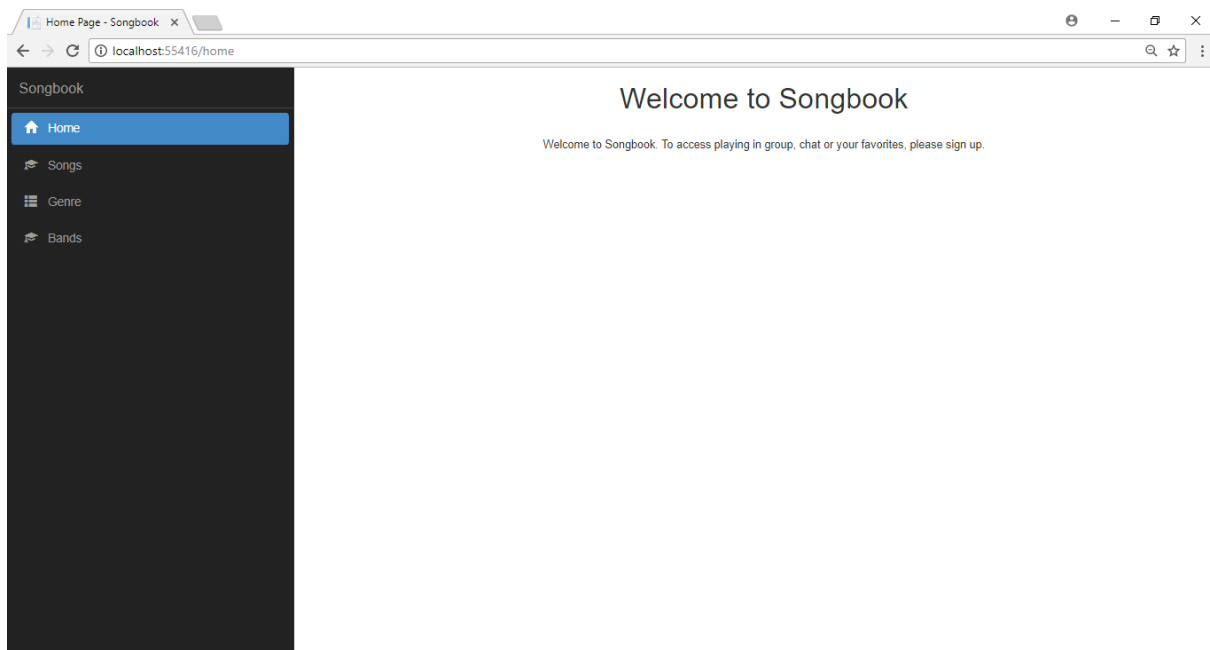
5.1 Základní popis

Při vstupu do IS má uživatel možnost se přihlásit nebo pokračovat bez přihlášení, podle toho se pak liší jak vzhled, tak služby nabízené IS.



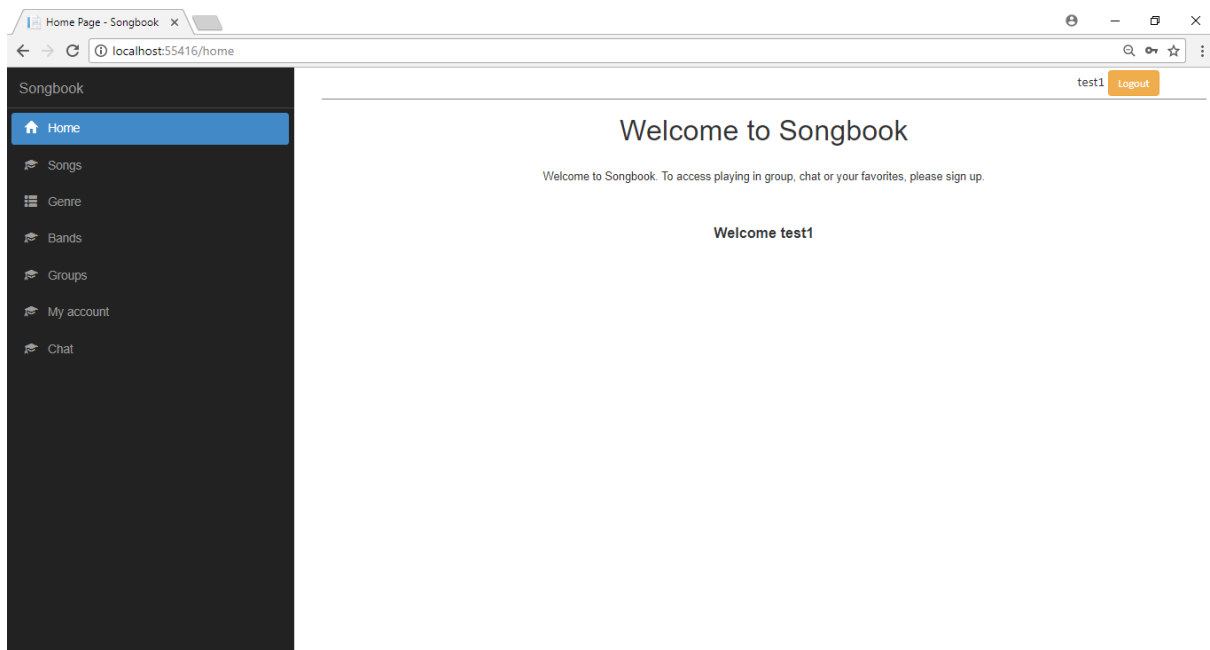
Obrázek 9: Zobrazení stránky - Přihlášení a registrace

Služby nabízené IS se liší podle toho, zda je uživatel přihlášen nebo nepřihlášen. Mimo služby se také mění vzhled aplikace. Na levé straně je umístěno menu. Pro nepřihlášeného uživatele obsahuje pouze 4 položky, přes které je schopen se dostat k zobrazení písní. A to buď přes seznam písní, přes umělce, nebo přes žánr. Menu uživatele je rozšířeno o další 3 položky. Pomocí položky group se uživatel přesune k výpisu skupin, kde má možnost vytvořit novou skupinu nebo zobrazit detail existující. V tomto detailu může poslat žádost o přijetí, kterou musí schválit zakladatel skupiny. Pokud je členem skupiny, může se účastnit synchronizovaného skupinového hraní. Položka My account umožňuje uživateli přístup k písním, jenž sám vložil do IS a jejich úpravu, či odstranění. Také je tam umístěn výpis jeho oblíbených písní a tyto písně si může buď zobrazit, nebo odebrat ze seznamu. Chat slouží ke komunikaci všech uživatelů využívajících IS.



Obrázek 10: Zobrazení stránky - Nepřihlášený uživatel

Nepřihlášený uživatel je omezen ve využití služeb, tudíž i design je jednodušší. Chybí horní lišta se jménem uživatelem a tlačítkem pro přihlášení. Nepřihlášení uživatelé mohou pouze zobrazovat uložená data a z toho důvodu design pro nepřihlášené uživatele obsahuje mnohem méně tlačítek.



Obrázek 11: Zobrazení stránky - Přihlášený uživatel

5.2 Autentizace a správa uživatelů

Přihlášení a registrace bude probíhat přes vlastní přihlašování. Nebude tedy k němu využíváno žádné API. Pro vytvoření nového účtu bude potřeba, aby uživatel zadal přihlašovací jméno, které ještě není uloženo v systému (atribut E-mail v tabulce User bude sloužit pro uložení přihlašovacího jména a bude nastaven na UNIQUE). U každého vkládání, kromě hodnocení, je zobrazeno přihlašovací jméno uživatele. Ať už komunikuje skrze chat, přidává komentáře, či vkládá písničky.

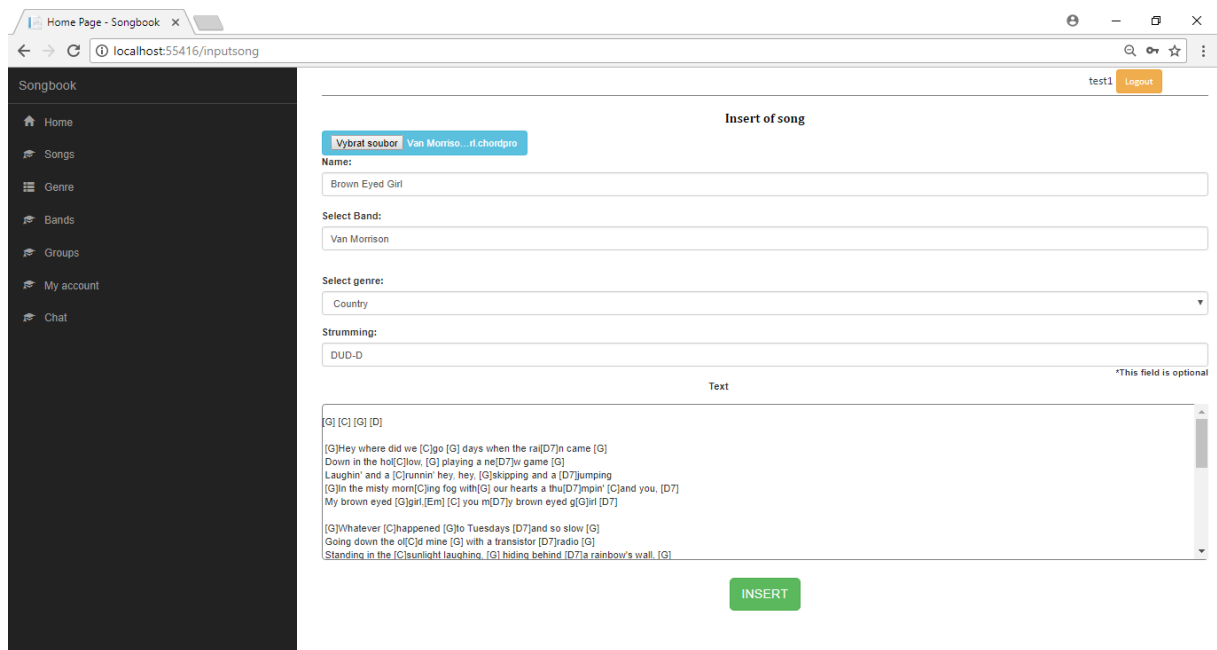
Po zadání přihlašovacích dat je heslo zašifrováno. Pro šifrování hesel bude využit balíček ts-md5, který slouží k šifrování řetězců i souborů. Takto upravené heslo se uloží do databáze.

```
var hashed = Md5.hashStr(pwd);
```

Výpis 3: Ukázka šifrování hesla

5.3 Vložení nové písničky

Pro vložení nové písničky do IS bude muset být uživatel přihlášen. K vložení bude mít dvě možnosti. Buď vloží písničku vypsáním požadovaných informací, nebo je nahraje ze souboru a chybějící informace doplní. IS bude umožňovat načtení dat ze dvou typů souborů. Prvním je textový soubor (.txt) a druhým je soubor typu chordpro (.chordpro). Mezi požadované informace patří název písničky, jméno umělce, do jakého žánru písnička patří a text písničky. Rytmus/strumming je volitelná informace, která se v případě nevyplnění nastaví na pomlčku. ID uživatele, jenž písničku vložil, je ukládáno automaticky. Text je vkládán ve formátu chordpro, tzn. akordy jsou napsány v hranatých závorkách před znakem, nad kterým mají být zobrazeny. Informace o autorovi, umělci atd. jsou ve složených závorkách.



Obrázek 12: Zobrazení stránky - Přidání písně

Při načtení informací ze souboru je potřeba rozdělit data, které soubor obsahuje. K tomu je využita následující funkce.

```
public onFileSelected(event: any) {
    var reader = new FileReader();
    reader.readAsText(event.srcElement.files[0]);
    var me = this;
    reader.onload = (e) => {
        me.fileText = reader.result;
        var text = "";
        for (let i = 0; i < me.fileText.length; i++) {
            if (me.fileText[i] == '{') {
                var veta = "";
                var title = "";
                var half = false;
                i = i + 1;
                while (me.fileText[i] != '}') {
                    if (me.fileText[i] == ':') {
                        half = true;
                        i = i + 1;
                    }
                    if (half == false) {title = title + me.fileText[i];}
                }
            }
        }
    }
}
```

```

        if (half == true) {veta = veta + me.fileText[i];}
        i++;
    }
    i++;
    if (title == "title" || title == "t") {this.namesong = veta;}
    if (title == "subtitle" || title == "st" || title == "artist"
        ) {this.bandvalue = veta;}
    }
    text = text + me.fileText[i];
}
this.text = text;
this.text = this.text.trim();
}
}

```

Výpis 4: Rozložení dat ze souboru při vkládání písně

Funkce načte obsah souboru jako jeden řetězec. Ten je procházen cyklem for a pokud je znak, na kterém je současná iterace, roven znaku {, tak víme, že začínají informace o písni (kromě textu, jenž není ve složených závorkách). Cyklem while procházíme text dále, dokud nenarazíme na pravou závorku. Vevnitř cyklu while rozdělujeme dvojtečkou název informace a její hodnotu. Když už text neobsahuje složené závorky, víme, že další část řetězce je už text. Na ten zavoláme funkci trim. Tím zajistíme odstranění prázdných znaků na začátku a konci písně. Po rozložení řetězce jsou informace vloženy do vkládacích oken, aby je mohl uživatel dále upravit.

Pro vložení písně zavoláme uloženou proceduru v databázi. První ověříme, zda daná píseň od daného uživatele již není uložena a pokud je, tak tím procedura končí. Pokud se taková píseň ještě nevyskytuje v databázi, tak procedura pokračuje. Další podmínkou ověříme, zda je v databázi uložen umělec dané písně. Pokud existuje, tak uložíme jeho ID do pomocné proměnné. Pokud umělec neexistuje, tak je vložen do databáze a ID nově vloženého umělce se uloží do pomocné proměnné. Dále vložíme hodnoty do tabulek Band_Songs, Songs_genre a Evaluation. V tabulce Evaluation vytvoříme záznam s ID uživatele, jenž vložil píseň a s ID vložené písně. Atribut value je nastavený na null.

5.4 Vizualizace písně

Jelikož text písně je uložen i s akordy v jednom řetězci, je potřeba tento řetězec rozeřít na akordy a text. K tomu jsou vytvořeny 3 navazující metody, pro přehlednější a snadnější porozumění.

```
public void Divider(string text)
```

```

{
    List<string> chords = new List<string>();
    List<string> lyrics = new List<string>();
    int j;

    for (int i = 0; i < text.Length; i++)
    {
        if (text[i] == '[')
        {
            text = text.Remove(i, 1);
            j = i;
            string oneChord = "";
            while (text[j] != ']')
            {
                oneChord = oneChord + text[j].ToString();
                text = text.Remove(j, 1);
            }
            if (text[i] == ']') { text = text.Remove(i, 1); }
            chords.Add(oneChord);
        }
        else
        {
            chords.Add("&nbsp;");
        }
        lyrics.Add(text[i].ToString());
    }
    Devider2(lyrics, chords);
}

```

Výpis 5: Dělení textu - první část

V této části kódu jsme rozdělili řetězec. Do jednoho pole jsme vložili akordy a do druhého text. Zároveň jsme odstranili závorky. Abychom dosáhli umístění akordu nad pozici, kde byl vložen, tak v každé iteraci, která se netýká akordů, vložíme do pole akordů pevnou mezeru.

```

public void Devider2(List<string> lyrics, List<string> chords)
{
    List<int> spaces = new List<int>();

```

```

int countOfSpaces = 1;
for (int i = 0; i < lyrics.Count; i++)
{
    if (lyrics[i] == " ")
    {
        if ((countOfSpaces % 5) == 0) { spaces.Add(i); }
        countOfSpaces++;
    }
    if (i == lyrics.Count - 1)
    {
        spaces.Add(i);
    }
    Devider3(lyrics, chords, spaces);
}
}

```

Výpis 6: Dělení textu - druhá část

Uložení pozice každé 5. mezery v textu. Jelikož se text neukládá s řádkováním, tak bude řádkován po každé 5. mezeře.

```

public void Devider3(List<string> lyrics, List<string> chords, List<int> spaces)
{
    int enter = 0;
    int j, forStart, forEnd;
    forStart = 0;
    textToShow = "";
    for (int i = 0; i < chords.Count; i++)
    {
        if (i == spaces[enter])
        {
            forEnd = spaces[enter];
            for (int st = forStart; st <= forEnd; st++)
            {
                textToShow = textToShow + chords[st].ToString();
            }
            textToShow = textToShow + " <br /> ";
            for (int st = forStart; st <= forEnd; st++)
            {

```



```

        textToShow = textToShow + lyrics[st].ToString();
    }
    forStart = forEnd;
    textToShow = textToShow + " <br /> ";
    enter++;
}
}
}

```

Výpis 7: Dělení textu - třetí část

Vložíme akordy a text do řetězce pod sebe s použitím HTML5 tagů na odřádkování. V HTML stránce proto musíme využít pro span, ve kterém je text zobrazován, [InnerHTML] místo [(ng-Model)]. InnerHTML se postará, ať se nám řetězec nezobrazí jako normální text, ale je brán jako část HTML stránky.

```
<span [innerHTML]=song.text></span>
```

Výpis 8: InnerHTML

Pro automatický posun obrazovky použijeme balíček @nickylenaers/ngx-scroll-to, který nám umožňuje nastavit dobu scrolování. Tu přepočítáváme z hodnot zadaných uživatelem na posuvníku. Rozdíl mezi jednotlivými políčky je 4000 milisekund.

```

public triggerScrollTo() {

    const config: ScrollToConfigOptions = {
        target: 'anchor',
        duration: this.speed
    };
    this._scrollToService.scrollTo(config);
}

```

Výpis 9: Automatický posun obrazovky

5.5 Synchronizované skupinové hraní

Funkce synchronizovaného skupinového hraní využívá .NET knihovny SignalR, která umožňuje komunikaci v reálném čase. Pro využití této funkce, musí být uživatel přihlášen a být členem

nebo zakladatelem skupiny. Pokud je uživatel zakladatelem skupiny, má možnost zahájit synchronizované hraní, ke kterému se poté přidají členové skupiny, kteří chtějí hrát. Pokud uživatel není členem skupiny, tak se synchronizovaného hraní účastnit nemůže. Dále zakladatel vybere píseň, jenž se bude hrát . Po vybrání se daná píseň zobrazí všem připojeným členům. U zobrazené písně může zakladatel určit rychlost automatického posunu obrazovky, která bude stejná u všech připojených uživatelů. Zakladatel má možnost výběru další písně, zatímco členové zůstávají na předchozí písni. Ke komunikaci během skupinového hraní slouží chat, jenž je umístěn ve spodní části písně.

```

public void CreateRoom(int roomName)
{
    string key = roomName.ToString();
    var group = _groups.FirstOrDefault(g => g.Key == key);
    if (group == null)
    {
        group = new PlayingGroups { Key=key, Owner = Context.ConnectionId,
            Created = true };
        _groups.Add(group);
        group.Members.Add(new Memb { ConnectionId = Context.ConnectionId });
    }
    else
    {
        group.Owner = Context.ConnectionId;
        group.Members.Add(new Memb { ConnectionId = Context.ConnectionId });
    }
    BroadcastGroup(group);
}

```

Výpis 10: Vytvoření skupiny pro synchronizované hraní

Metoda CreateRoom je volaná z Angular4 typescriptového souboru.

```

this.hubConnection.invoke('CreateRoom', this.group.gRid);

```

Výpis 11: Zavolání funkce pro vytvoření SignalR skupiny

Pouze zakladatel skupiny (tzn. skupiny uložené v databázi) může vytvořit skupinu (v SignalR). Název skupiny je předáván jako parametr. Ten je při volání funkce nastaven na ID skupiny, čímž se zabrání pokusu o vznik dvou skupin se stejným jménem. Pokud skupina je již vytvořena (a uživatel ji neodstranil zastavením synchronizovaného hraní), pouze se nastaví jeho ConnectionId na vlastníka skupiny.

```

this.hubConnection.invoke('JoinRoom', this.group.gRid);

```

Výpis 12: Zavolání funkce pro připojení k SignalR skupině

Pomocí tohoto zavolání funkce se mohou členové přidat ke skupinovému hraní.

```
public void JoinRoom(int roomName)
{
    string key = roomName.ToString();
    var group = _groups.FirstOrDefault(g => g.Key == key);
    if(group != null)
    {
        group.Members.Add(new Memb{ConnectionId = Context.ConnectionId});
        BroadcastGroup(group);
    }
}
```

Výpis 13: Připojení ke skupině pro synchronizované hraní

Pokud skupina s daným názvem existuje, pak je uživatel přidán do pole uživatelů dané skupiny. Zakladatel po výběru písně, či vybrání rychlosti automatického posunu posílá zprávu všem připojeným členům.

```
private void BroadcastGroup(PlayingGroups group)
{
    var clients = group.Members.Select(g => g.ConnectionId).ToList();
    Clients.Clients(clients).SendAsync("Group", group);
}
```

Výpis 14: Posílání zpráv všem členům skupiny

V typescript souboru je nastaven listener pro tyto zprávy.

```
ngOnInit()
{
    this.hubConnection = new HubConnection("/echo");
    this.hubConnection.on('Group', (data: SynchData) => {
        this.sdata = data;
    });

    this.hubConnection.start()
```

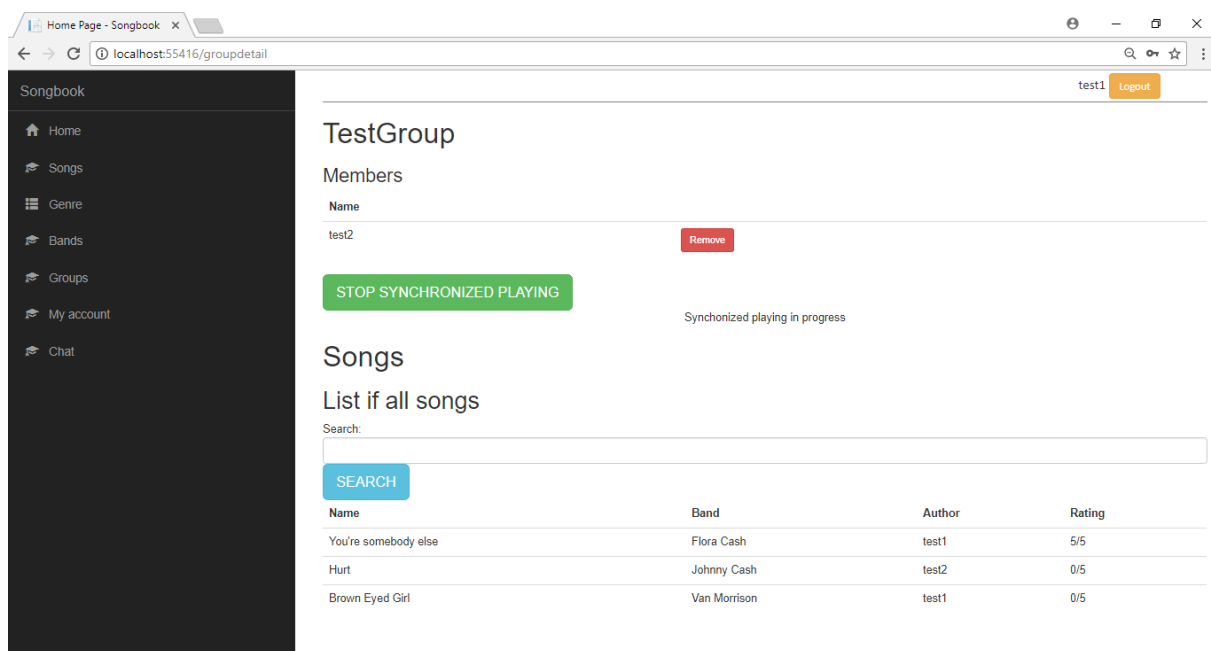
```

    .then(() => { console.log("Connection started"); })
    .catch(err => { console.error(err); });
}

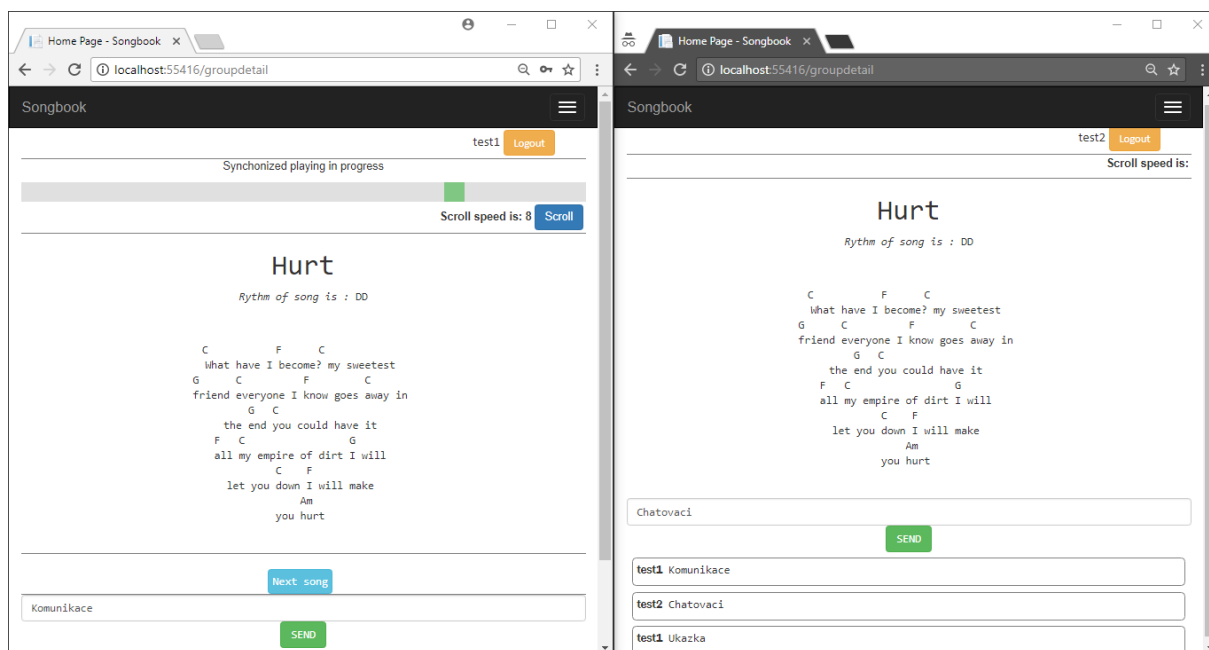
```

Výpis 15: Listener na zprávy v synchronizovaném hraní

Pomocí získaných dat se jim zobrazí daná píseň nebo spustí funkce pro automatický posun obrazovky.



Obrázek 13: Zobrazení stránky - Výběr písně pro synchronizované hraní



Obrázek 14: Zobrazení stránky - Ukázka zobrazené písně a chatu v synchronizovaném hraní

Na obrázku můžeme vidět rozdíl mezi oknem zakladatele a člena skupiny. Zakladatel má pro všechny výše zmíněné činnosti viditelné tlačítka, zatímco člen skupiny ne.

6 Porovnání s existujícími systémy

- Ultimate Guitar [26] je jeden z nejznámějších a nejobsáhlejších elektronických zpěvníků. Tento IS umožňuje uživateli vyhledávat písně dle žánru, autora, názvu písně, dekády, ze které píseň je, nebo z dynamického seznamu nejčastěji navštívených písní. Při zobrazení jedné písně umožňuje automatické posouvání obrazovky s nastavitelnou rychlostí.

Narozdíl od mnou vyvíjeného elektronického zpěvníku, tento zpěvník má možnost grafického zobrazení akordu při najetí kurzoru na jeho značení. Jako jednu z placených služeb (s měsíční trial verzí) nabízí učební program. V učebním programu jsou zobrazeny struny s místy, kde mají být drženy. Čímž ukazují uživateli, co má zrovna hrát, a to i bez znalosti akordů. Výhodou mého elektronického systému je možnost skupinového synchronizovaného hraní, který tento elektronický zpěvník neumožňuje. Ultimate Guitar obsahuje také fórum, pro komunikaci uživatelů a dotazování na funkčnost IS.

Přestože je tento IS velmi moderní, nemá responzivní design. Čímž je limitován typ zařízení, na kterých je možné IS využívat.

- E-Chords [27] je dalším ze známých a velmi propracovaných elektronických zpěvníků. Mezi jeho největší výhody patří možnost uložení několika verzí jedné písně pro různé nástroje. Mezi ně patří kytara, ukulele, bubny, keyboard, flétna, harmonika a bassová kytara. U zobrazení jedné písně umožňuje automatický posun textu, avšak tento posun není plynulý ale trhavý. Jednou z vychytávek je možnost zvětšení nebo zmenšení písma, nebo vyhledávání podle části textu.

Stejně jako Ultimate Guitar [26] má tento zpěvník možnost grafického zobrazení akordu při najetí kurzoru na jeho značení. Umožňuje také lekce, které jsou provedeny pomocí video tutoriálu. Umožňuje vkládání písní s určením akordů pomocí tagu <R> pro zahájení a </R> pro ukončení tagu.

Další z mála chyb tohoto IS je absence responzivnosti designu. Pro některé uživatele může být nepříjemný rozdíl mezi placenou a neplacenou verzí. Například písně pro keyboard jsou přístupné pouze pro uživatele využívající placenou verzi. Výhodou je, že prohlížet písně lze i bez registrace.

- Heart Wood Guitar [28] je méně známý než předchozí dva zpěvníky a ne tak propracovaný. Chybí zde vyhledávání v písních, automatický posun textu, vyhledání podle žánru nebo umělce. Neobsahuje hodnocení, či komentování písní, ani možnost grafického zobrazení akordu.

Na rozdíl od předchozích dvou IS, tento má responzivní design. Také umožňuje uložit píseň do textového souboru, ale neumí píseň ze souboru načíst.

- Zpěvník s akordy [29] je poslední zpěvník, který budu porovnávat. Jedná se o český elektronický zpěvník, který má ze všech výše zmíněných nejnížší počet služeb. Vyhledávání

je jednorázové, po zadání do vyhledávače jsme odkázáni na další stránku a pro další vyhledání se musíme vracet. Písňe může vložit kdokoliv, a to i bez registrace. Neumožňuje automatický posun, grafické zobrazení akordu, ani skupinovou synchronizaci. Také neobsahuje našeptávač, a to v žádném vyhledávání.

Neobsahuje responzivní design, ani neumí načíst informace ze souboru.

7 Závěr

Cílem bakalářské práce bylo vytvořit elektronický zpěvník, který by využíval nové technologie a byl tak uživatelsky přívětivější.

Stejně jako stávající elektronické zpěvníky i můj elektronický zpěvník umožňuje vizualizaci textu a akordů písní, jejich hodnocení a komentování. Některé zpěvníky nabízejí i možnost automatického posunu obrazovky a vkládání nových písní do IS. Můj zpěvník nabízí všechny tyto služby a navíc možnosti získané díky použití nových technologií.

Mezi tyto možnosti patří responzivní design, umožňující zobrazení stránky na jakémkoliv elektronickém zařízení, což ocení zejména lidé, kteří často cestují a nechtějí s sebou přenášet větší elektronické zařízení. Další z možností je vložení souboru obsahujícího píseň do IS. Tento soubor může být buď ve formátu ChordPro nebo v textovém souboru, čímž usnadňuje uživateli vkládání nových písní. Na rozdíl od všech mnou nalezených elektronických zpěvníků, můj zpěvník nabízí synchronizované skupinové hraní. Díky této službě, může hrát skupina lidí na více zařízeních současně stejnou píseň, se stejnou rychlostí a stejným momentem zahájení automatického posunu obrazovky. Všechny služby, které nejsou běžné u stávajících elektronických zpěvníků, jsou podrobně popsány v kapitole Implementace.

Během implementace jsem se snažil co nejvíce využít správných návrhových vzorů, aby byl IS snadno rozšiřitelný. Jsem rád, že jsem se tímto způsobem mohl naučit, jak pracovat s technologií Angular 4 a především s knihovnou SignalR.

Literatura

- [1] ABC Music Notation. URL: http://trillian.mit.edu/~jc/music/abc/doc/ABCtut_Chords.html
- [2] JAM Notation. URL: <http://pyva.net/eng/jam/>
- [3] Lily Pond. URL: <https://libremusicproduction.com/tools/lilypond>
- [4] ChordPro. URL: <https://www.chordpro.org/chordpro/index.html>
- [5] Martin Fowler - Návrhové vzory. URL: <https://martinfowler.com/>
- [6] Data Mapper. URL: <https://martinfowler.com/eaCatalog/dataMapper.html>
- [7] Table Data Gateway. URL: <https://martinfowler.com/eaCatalog/tableDataGateway.html>
- [8] MS SQL. URL: <https://searchsqlserver.techtarget.com/definition/SQL-Server>
- [9] T-SQL. URL: https://www.tutorialspoint.com/t_sql/index.htm
- [10] .NET Core 2.0. URL: <https://www.c-sharpcorner.com/article/what-is-new-in-net-core-2-0/>
- [11] Serializace URL: <https://www.itnetwork.cz/csharp/soubory/tutorial-csharp-serializace-a-deserializace>
- [12] Socket URL: <https://docs.microsoft.com/en-us/dotnet/framework/network-programming/sockets>
- [13] HTTP URL: [https://msdn.microsoft.com/cs-cz/library/system.net.http.httpclient\(v=vs.110\).aspx](https://msdn.microsoft.com/cs-cz/library/system.net.http.httpclient(v=vs.110).aspx)
- [14] WebSocket URL: [https://msdn.microsoft.com/en-us/library/system.net.websockets.websocket\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.net.websockets.websocket(v=vs.110).aspx)
- [15] XPath URL: [https://msdn.microsoft.com/cs-cz/library/ms256086\(v=vs.110\).aspx](https://msdn.microsoft.com/cs-cz/library/ms256086(v=vs.110).aspx)
- [16] XQuery URL: https://developers.de/blogs/damir_dobric/archive/2007/12/13/xquery-examples.aspx
- [17] SignalR. URL: <https://docs.microsoft.com/cs-cz/aspnet/signalr/overview/getting-started/introduction-to-signalr>
- [18] Forever Frame. URL: [https://msdn.microsoft.com/en-us/library/microsoft.aspnet.signalr.transports.foreverframetransport\(v=vs.118\).aspx](https://msdn.microsoft.com/en-us/library/microsoft.aspnet.signalr.transports.foreverframetransport(v=vs.118).aspx)

- [19] Server Send Events. URL: <https://www.c-sharpcorner.com/blogs/server-side-events-in-asp-net-mvc>
- [20] Long Polling. URL: <https://techoctave.com/c7/posts/60-simple-long-polling-example-with-javascript-and-jquery>
- [21] HTML5. URL: <http://www.html5.cz/>
- [22] Angular 4. URL: <https://angular.io/docs>
- [23] Angular 4. URL: https://www.tutorialspoint.com/angular4/angular4_overview.html
- [24] Angular 4 - NPM. URL: <https://www.npmjs.com/>
- [25] Bootstrap. URL: <https://getbootstrap.com/>
- [26] Ultimate Guitar. URL: <https://www.ultimate-guitar.com/>
- [27] E-Chords. URL: <https://www.e-chords.com/>
- [28] Heart Wood Guitar. URL: <http://www.heartwoodguitar.com/>
- [29] Bootstrap. URL: <http://www.zpevnik.wz.cz>